WHAT IS CLAIMED IS:

1. A multiple priority non-blocking load buffer comprising:

a plurality of variable depth pending queues for queuing entries of memory or I/O requests generated by processors to peripheral devices, each of said
5   pending queues including a plurality of sub-queues; and

a variable length return queuing unit for buffering data returned from said peripheral devices in response to said I/O requests.

2. A multiple priority non-blocking load buffer according to Claim 1, wherein said variable length return queuing unit comprises one variable length
10   return queue.

3. A multiple priority non-blocking load buffer according to Claim 1, wherein said variable length return queuing unit comprises a plurality of variable length return queues.

4. A multiple priority non-blocking load buffer according to Claim 3,
15   wherein each of said plurality of variable length return queues corresponds to each of said processors.

5. A multiple priority non-blocking load buffer according to Claim 3, wherein each of said plurality of variable length return queues corresponds to each of said peripheral devices.

20   6. A multiple priority non-blocking load buffer according to Claim 3, wherein each of said plurality of variable length return queues corresponds to a unique priority level.

7.   A multiple priority non-blocking load buffer according to Claim 1, wherein each of said sub-queues within one of said pending queues is assigned a unique priority level.

8.   A multiple priority non-blocking load buffer according to Claim 7,
5   wherein said I/O requests include an address and a priority tag, said address directs said memory or I/O requests to a corresponding one of said pending queues and said priority tag channels said memory or I/O requests to a corresponding one of said sub-queues within the one said pending queue.

9.   A multiple priority non-blocking load buffer according to Claim 8,
10   wherein each of said pending queues comprises a priority controller for issuing said memory or I/O requests from said sub-queues in a highest priority first manner.

10.   A multiple priority non-blocking load buffer comprising:

a variable depth pending queue for queuing entries of I/O requests generated by a processor to peripheral devices, said pending queue including a
15   plurality of sub-queues; and

a variable length return queuing unit for buffering data returned from said peripheral devices in response to said memory or I/O requests.

11.   A multiple priority non-blocking load buffer according to Claim 10, wherein said variable length return queuing unit comprises one variable length
20   return queue.

12.   A multiple priority non-blocking load buffer according to Claim 10, wherein said variable length return queuing unit comprises a plurality of variable length return queues.

13. A multiple priority non-blocking load buffer according to Claim 12, wherein each of said plurality of variable length return queues corresponds to each of said peripheral devices.

14

~~14.~~ A multiple priority non-blocking load buffer according to Claim 12,

5   wherein each of said plurality of variable length return queues corresponds to a unique priority level.

15. A multiple priority non-blocking load buffer according to Claim 10, wherein each of said sub-queues within said pending queues is assigned a unique priority level.

10   16. A multiple priority non-blocking load buffer according to Claim 15, wherein said memory or I/O requests include a priority tag, said priority tag channels said memory or I/O requests to a corresponding one of said sub-queues.

16   *15*

~~17.~~ A multiple priority non-blocking load buffer according to Claim ~~16,~~ wherein said pending queue comprises a priority controller for issuing said memory

15   or I/O requests from said sub-queues in a highest priority first manner.

18. A method for limiting admission of prioritized memory or I/O transactions for storage in a memory comprising:

(a) dividing the memory into a plurality of queues having unique priority levels;

20   (b) channeling said prioritized memory or I/O transactions to the ones of said queues having its unique priority level corresponding to said prioritized memory or I/O transactions;

(c) storing said prioritized memory or I/O transactions in the corresponding queues when space is available therein; and

(d) storing said prioritized memory or I/O transactions in one of said queues having a priority level which is one priority level less than the unique priority level corresponding to said prioritized memory or I/O transactions channeled at said step (b) when the space available in the queues corresponding to said prioritized memory or I/O transactions is exceeded.

19.   A method according to Claim 18, wherein the memory is contained within a non-blocking load buffer.

20.   A method according to Claim 18, wherein said memory or I/O transactions comprise loads and stores.